Algebraic cryptanalysis of HFE using Gröbner bases

Jean-Charles Faugère

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Algebraic cryptanalysis of HFE using Gröbner bases

Jean-Charles Faugère

N° 4738

Février 2003

THÈME 2

ISSN 0249-6399 ISRN INRIA/RT--0276--FR+ENG

INRIA
LORRAINE

Algebraic cryptanalysis of HFE using Gröbner bases

Jean-Charles Faugère*

Thème 2 — Génie logiciel
et calcul symbolique
Projets SPACES

Rapport de recherche n° 4738 — Février 2003 — 19 pages

Abstract: HFE (Hidden Fields Equations) is a public key cryptosystem using (multivariate) polynomial operations over finite fields. It has been proposed by Jacques Patarin following the ideas of Matsumoto and Imai. In this paper we present a new and efficient attack of this cryptosystem based on fast algorithms for computing Gröbner basis. The attack consists simply in computing a Gröbner basis of the public key. Of course the efficiency of this attack depends strongly on the choice of the algorithm for computing the Gröbner basis: while the corresponding algebraic systems are completely far beyond the capacity of any implementation of the Buchberger algorithm, it was was possible to break the first HFE challenge (80 bits) in only two days of CPU time by using the new algorithm F5 implemented in C. We establish experimentally that the algebraic systems coming from HFE behave *not as* "random systems" so that they can be solved in *polynomial time* when the degree d of the univariate polynomial is fixed. For practical value of d we can establish precisely the complexity of this attack: $O(n^8)$ (resp. $O(n^{10})$) when 16<d<128 (resp. 128<d<513).

Key-words: Hidden Field Equations (HFE), Multivariate polynomial equations, Gröbner bases, Algebraic Cryptanalysis, Computer Algebra.

* Projet SPACES LIP6/LORIA CNRS/UPMC/INRIA

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,
615, rue du Jardin Botanique, BP 101, 54602 Villers-Lès-Nancy (France)
Téléphone : +33 3 83 59 30 00 — Télécopie : +33 3 83 27 83 19

Cryptanalyse algèbrique de HFE par les bases de Gröbner

Résumé : HFE (Hidden Fields Equations) est un cryptosystème à clé publique basé sur les polynômes (multivariés) dans les corps fini. HFE a été proposé par Jacques Patarin en suivant les idées de Matsumoto et Imai. Dans cet article nous présentons une nouvelle attaque très efficace basée sur les nouveaux algorithmes de calcul des bases de Gröbner. L'attaque consiste simplement à calculer une base de Gröbner de la clé publique. Bien sur l'efficacité de cette attaque dépend fortement du choix de l'algorithme utilisé pour calculer la base de Gröbner: alors que les systèmes algébriques provenant de HFE sont complètement inaccessibles aux meilleures implantations de l'algorithme de Buchberger, il est possible de résoudre le premier challenge HFE (80 bits) en seulement deux jours de temps CPU en utilisant le nouvel algorithme F5 (implanté en C). Nous établissons expérimentalement que les systèmes algébriques issus de HFE *ne* se comporte *pas* comme des systèmes *aléatoires* et qu'ils peuvent être résolu en temps polynômial lorsque le degré d du polynôme secret est fixé Plus exactement, et pour les valeurs admissibles de d, on montre que la complexité de l'attaque est : $O(n^8)$ (resp. $O(n^{10})$) quand 16<d<128 (resp. 128<d<513).

Mots-clés : Polynômes multivariés, Bases de Gröbner, Cryptanalyse algébrique, Calcul Formel.

1 Introduction

The security of many public key cryptosystems relies on the intractability of some well known mathematical problem (integer factorization, ...). Since solving system of algebraic equations is a difficult problem (NP-complete), it is a good candidate for the design of new public key encryption and signature schemes. HFE (Hidden Fields Equations) is a public key cryptosystem using (multivariate) polynomial operations over finite fields. It has been proposed by Jacques Patarin [Pat96b] following the ideas of Matsumoto and Imai [MI88]. It has long been regarded as a very promising cryptosystem because it can be used to produce signatures as short as 128, 100 and even 80 bits.

In [KS99] a polynomial time attack on HFE was presented; this method is based on "relinearization" techniques. In [Cou01] the complexity of this attack was estimated to be at last $n^{\log^2 d}$ where d is the degree of the (secret) univariate polynomial of HFE. The same attack was improved by Courtois[Cou01] to obtain a theoretical complexity of $n^{3\log_2 d+\mathcal{O}(1)}$.

In this paper we present a new and efficient attack of this cryptosystem based on fast algorithms for computing Gröbner basis [Buc65, Buc70, Buc79]. The public key of HFE is a list of algebraic equations, and since Gröbner bases is a well known and efficient method for solving polynomial system of equations, our attack is very simple: we compute a Gröbner basis of the public key of HFE. Of course the efficiency of this attack depends strongly on the choice of the algorithm for computing the Gröbner basis. With the best implementation of the Buchberger algorithm [Buc65] only toys examples can be solved ($\approx$ 20 bits). On the other hand, by using the new and efficient F_5 ([Fau02]) algorithm for computing Gröbner we were able to break the first HFE challenge (80 bits) in only two days of CPU time on a single processor (Alpha). The goal of this paper is to present a methodology to study experimentally a cryptosystem like HFE with algebraic tools. We made a series of computer simulations on real size HFE problems (up to 160 bits) so that we can establish precisely the complexity of the Gröbner attack: for all practical values of d (less than 512) the complexity is at most $\mathcal{O}(n^{10})$.

In [CSPK00], another algorithm (XL) was proposed for solving algebraic systems over finite fields. It is clearly an interesting point to compare the XL and the Gröbner approaches; but the evaluation of the theoretical complexity of such algorithms is difficult. Moreover, as many other algorithms (LLL, the simplex algorithm for solving linear programs, ...), Gröbner bases algorithms behave much better in practice than in the worst case, so considering just the worst-case bounds may lead to underestimate their practical utility. A typical example is precisely the computation of Gröbner bases over $\mathbb{F}_2$, whose asymptotic worst-case time bound is exponential, while its running time is bounded by a low-degree polynomial for HFE. No benchmarks or implementation of the algorithm XL are available so the comparison with XL is out of the scope of this paper and is the subject of another paper.

2 Description of HFE

We refer to [Pat96b] for a complete description of HFE and we describe "the basic HFE" (HFE without variations). We denote by $\mathbb{F}_2$ (resp. $\mathbb{F}_{2^n}$), the finite field of cardinality (2) (resp. 2^n) and characteristic 2. Let

$$f(x) = \sum \beta_{i,j} x^{2^{\theta_{i,j}}+2^{\varphi_{i,j}}} + \sum_k \alpha_k x^{2^{\epsilon_k}} + \mu$$

be a polynomial in x over $\mathbb{F}_{2^n}$ of degree d, for integers $\theta_{i,j}, \varphi_{i,j}, \epsilon_k \geq 0$. In the rest of this paper d is always the degree of the univariate polynomial f.

Since $\mathbb{F}_{2^n}$ is isomorphic to $\mathbb{F}_2[z]/(g(z))$ where $g(z) \in \mathbb{F}_2[z]$ is irreducible of degree n, elements of $\mathbb{F}_{2^n}$ may be represented as n-tuples over $\mathbb{F}_2$, and f may be represented as a polynomial in n variables $x_1, \ldots, x_n$ over $\mathbb{F}_2$:

$$f(x_1, \ldots, x_n) = (q_1(x_1, \ldots, x_n), \cdots, q_n(x_1, \ldots, x_n))$$

with $q_i(x_1, \ldots, x_n) \in \mathbb{F}_2[x_1, \ldots, x_n]$ for $i = 1, 2, \ldots, n$. The q_i are polynomials of *total degree* 2 due to the choice of f and the fact that $x \mapsto x^2$ is a linear function of $\mathbb{F}_{2^n} \longrightarrow \mathbb{F}_{2^n}$.

Let S and T be two $n \times n$ non singular matrices then we can compose S, f and T:

$$S(f(TX)) = (p_1(x_1, \ldots, x_n), \cdots, p_n(x_1, \ldots, x_n))$$

where $\mathbb{F}_2^n$ is regarded as an n-dimensional vector space over $\mathbb{F}_2$ and X is the vector $(x_1, \ldots, x_n)$. Obviously p_i are again quadratic polynomials.

We can now describe the HFE (Hidden Field Equations) public key encryption scheme:

Secret key. The function f , two affine bijections S and T as above.

Public key. Some way of representing $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$. Polynomials p_i for $i = 1, 2, \ldots, n$ as above, computed using the secret key f , S, T.

Encryption. To encrypt the n-tuple $x = (x_1, \ldots, x_n) \in (\mathbb{F}_2)^n$(representing the message), compute the ciphertext

$$y = (p_1(x_1, \ldots, x_n), \cdots, p_n(x_1, \ldots, x_n))$$

Decryption. To decrypt the ciphertext y, first find all solutions z to the univariate equation $f(z) = T^{-1}y$, next compute $S^{-1}z$.

When the polynomial f is a monomial the HFE reduces to cryptosystem of Matsumoto and Imai [MI88] broken by Patarin in [Pat95a]. In the following we consider only *random quadratic* polynomial ($\beta_{i,j}$, α_k, μ random in $\mathbb{F}_{2^n}$).

Find the roots of a polynomial of degree d with coefficients in $\mathbb{F}_{2^n}$ can be done (see [vzGG99] for instance) in $\mathcal{O}(\mathbf{M}(d)\log(d))$ operations if $\mathbb{F}_{2^n}$ where $\mathbf{M}(d)$ is the cost of polynomial multiplication. We report the time to find *one* solution of univariate polynomial with NTL[Sho03] (PC PIII 1000 Mhz):

(n, d)	(80,129)	(80,257)	(80,513)	(128,129)	(128,257)	(128,513)
NTL (CPU time)	0.6 sec	2.5 sec	6.4 sec	1.25 sec	3.1 sec	9.05 sec

From these experimental results we conclude that, in practice, we cannot take arbitrarily big value for the degree of the univariate polynomial (say $d \leq 512$). The recommended values [Pat96b, Pat96c] for n are $n \geq 32$ and $n = 80$, $d = 96$ for the first HFE Challenge.

3 Gröbner basis

3.1 Mathematical definition of Gröbner bases

We refer to [Bec93, CLO92] for basic definitions. Let k be a field ($\mathbb{F}_q$ a finite field for instance) and $R = k[x_1, \ldots, x_n]$ the ring of multivariate polynomials. To a system of equations

$$f_1(x_1, \ldots, x_n) = \cdots = f_m(x_1, \ldots, x_n) = 0$$

we associate the ideal I generated by $f_1, \ldots, f_m$. A monomial in $x_1, \ldots, x_n$ is a term in $x_1, \ldots, x_n$ and a coefficient. We choose $<$ an admissible ordering on the monomials in $x_1, \ldots, x_n$. For instance the lexicographical ordering is such that $x_1^{\alpha_1} \cdots x_n^{\alpha_n} < x_1^{\beta_1} \cdots x_n^{\beta_n}$ iff $\alpha_i = \beta_i$ for $i = 1, \ldots, k$ and $\alpha_k < \beta_k$ for some k. Next for each polynomial f in R we can define its leading term $\mathrm{LT}(f)$ (resp. its leading monomial $\mathrm{LM}(f)$) to be the biggest term (resp. monomial) with respect to $<$.

Definition 1 *G a finite set of elements of I is a Gröbner basis of $(f_1, \ldots, f_m)$ wrt $<$ if for all $f \in I$ there exists $g \in G$ such that $LT(g)$ divides $LT(f)$.*

Let K be a field containing k, we can define the set of solutions in K which is the algebraic variety:

$$V_K = \{(z_1, \ldots, z_n) \in K \quad | f_i(z_1, \ldots, z_n) = 0 \ i = 1, \ldots, m\}$$

which is in fact the set of roots of the system of equations. Gröbner bases can be used in various situation (for instance when the number of solution is infinite of for computing real solutions). In the case of HFE we want to compute solutions of algebraic systems in $\mathbb{F}_2$. The following proposition tell us how to use Gröbner bases in order to *solve* a system over $\mathbb{F}_2$:

Proposition 1 *The Gröbner basis of $[f_1, \ldots, f_m, x_1^2 - x_1, \ldots, x_n^2 - x_n]$, in $\mathbb{F}_2[x_1, \ldots, x_n]$, describe all the solutions of $V_{\mathbb{F}_2}$. Particular useful cases are:*

i) $V_{\mathbb{F}_2} = \emptyset$ (no solution) iff $G = [1]$.

2) $V_{\mathbb{F}_2}$ has exactly one solution iff $G = [x_1 - a_1, \ldots, x_n - a_n]$ where $a_i \in \mathbb{F}_2$. Then $(a_1, \ldots, a_n)$ is the solution in $\mathbb{F}_2$ of the algebaric system.

This proposition tell us that we have to add the "field equations" $x_i^2 = x_i$ to the list of equations that we want to solve. Consequently we have to compute a Gröbner basis of $m+n$ polynomials and n variables. In fact, the more equations you have the more able you are to compute a Gröbner basis.

3.2 Useful properties of Gröbner bases

Another order on monomials is the Degree Reverse lexicographical order or (**DRL** order). This order is less intuitive than the lexicographical order but it has been shown that the DRL ordering is the most efficient, in general, for computating Gröbner bases.

$x_1^{\alpha_1} \cdots x_n^{\alpha_n} >_{\mathrm{DRL}} x_1^{\beta_1} \cdots x_n^{\beta_n}$ iff $deg(x^\alpha) = \sum_{i=1}^n \alpha_i > deg(x^\beta)$ or $deg(x^\alpha) > deg(x^\beta)$ and, in $\alpha - \beta \in \mathbb{Z}^n$, the right-most nonzero entry is negative.

We have seen in proposition 1 that Gröbner bases are useful to solve a system but they can also be used to discover low degree relations:

Proposition 2 *If G is a Gröbner basis of an ideal I for $<_{\mathrm{DRL}}$ then G contains all the (independent) equations in I of lowest total degree.*

By computing Gröbner bases it is even possible to find *all* the algebraic relations among $f_1, \ldots f_m$ (see [CLO92] page 338 for a precise definition of the ideal of relations).

Proposition 3 *([CLO92] page 340) Fix a monomial order in $k[x_1, \ldots, x_n, y_1, \ldots, y_m]$ where any monomial involving one of the $x_1, \ldots, x_n$ is greater than all monomials in $k[y_1, \ldots, y_m]$ (lexicographical ordering for instance) and let G be the Gröbner basis for this ordering. Then $G \cap k[y_1, \ldots, y_m]$ describe all the relations among $f_1, \ldots, f_m$.*

By combining proposition 2 and 3 we can thus find the lowest relations among the f_i. This will enable us to describe and generalize in another way the original attack of Patarin (see section 5.2) for the Matsumoto Imai cryptosystem.

3.3 Algorithms for computing Gröbner bases

Notice that definition 1 does *not depend on a particular algorithm.* Due to space limitations it is impossible to describe in details all the algorithms for computing Gröbner bases so we report only recent improvements. Historically the first algorithm for computing Gröbner basis was presented by Buchberger [Buc65, Buc70, Buc79]. The Buchberger algorithm is a very practical algorithm and it is implemented in all Computer Algebra Systems (a non exhaustive list of efficient implementation is: Magma, Cocoa, Singular, Macaulay, Gb, ...); section 4.1 contains a comparison between them for the HFE problem. More recently more efficient algorithms for computing Gröbner have been proposed. The first one F_4 [Fau99] reduces the computation to a linear algebra problem (the link between solving algebraic and Gaussian elimination is very old ([Mac16, Laz83] or even Sylvester)). More precisely the algorithm F_4 incrementally construct matrices in degree 2, 3, ... D:

$$A_D = \begin{array}{c} \\ m_1 \times f_{i_1} \\ m_2 \times f_{i_2} \\ m_3 \times f_{i_3} \\ \cdots \end{array} \begin{array}{c} \overbrace{}^{momoms\ degree\ \leq\ D\ in\ x_1, \ldots, x_n} \\ \left(\begin{array}{c} \cdots \\ \cdots \\ \cdots \\ \cdots \end{array} \right) \end{array}$$

where $m_1, m_2, \ldots$ are monomials such that the total degree of $m_j f_{i_j}$ is less than D. The next step in the algorithm is to compute a row echelon of A_D using linear algebra techniques. It must be emphasized that the rows of A_D is a *small subset* of all the possible rows $\{mf_i \text{ s.t. } 1 \leq i \leq m \;\; m \text{ any monomial } deg(m) \leq D - deg(f_i)\}$.

A even more efficient algorithm F_5 [Fau02] is now available: the number of rows in the generated matrix A_D is minimal and the matrix is full rank (under some conditions see [Fau02]). For the special case of $\mathbb{F}_2$ we use, in fact, a special version of this algorithm (called $F_5/2$) that takes into account the action of the Frobenius $h^2 = h$. Of course the implementation of the linear algebra part uses a dedicated version for $\mathbb{F}_2$.

From a complexity point of view the two important parameters are: D the maximal degree occurring in the computation and the size N_D of the matrix A_D. Then the whole complexity is simply N_D^ω where $2 \leq \omega \leq 3$ is the cost of linear algebra.

3.4 Complexity of Gröbner bases

Complexity of Gröbner bases (and more generally polynomial system solving) is the subject of a huge number of papers. Adding the "field equations" $x_i^2 - x_i$ imply a simple geometry of the set of solutions: all the ideals are radicals (no multiple roots), zero dimensional (finite number of solutions). In fact it is easy to prove:

Proposition 4 *The maximal degree D of the polynomials occurring in the computation of a Gröbner basis including field equations $x_i^2 = x_i$ is less than n. The complexity of the whole computation is bounded by a polynomial in 2^n.*

Remark 1 *Note that this result is only a rough upper bound. This must be compared with the complexity of the exhaustive search $\mathcal{O}(n2^n)$. In practice, however, efficient algorithms for computing Gröbner bases behave much better than in the worst case.*

A crucial point in the cryptanalysis of HFE is the ability to distinguish a "random" (or generic) algebraic system from an algebraic system coming from HFE. We will establish in section 4.2 that this can be done by computing Gröbner bases and comparing the maximal degree occurring in these computations. As a consequence we have to describe theoretically the behavior of such a computation. This study is beyond the scope of this paper and is the subject of another paper [BFS03] from which we extract some results. First the asymptotic behavior of the maximal degree occurring in the computation is:

$$d = \max \text{ total degree} \approx \frac{n}{11.114\ldots}$$

From this result we know that computing Gröbner bases of random systems is simply exponential; consequently, in practice, it is impossible to solve a system of n equations of degree 2 in n variables when n is big (say $n \geq 80$). From a practical point of view it is even more important to have *exact values* (see [BFS03]) for D and N_D when n is small:

n	14	15	16	17	$\cdots$	23	24	25	$\cdots$	80
degree	4	4	5	5	$\cdots$	5	6	6	$\cdots$	12
nb of rows	1695	1379	8840	11424	$\cdots$	40480	223124	278875	$\cdots$	$73526787216476 \approx 2^{46}$

Table 1. Maximal degree occurring in Gröbner for *random* systems.

For instance when $n = 80$, we read the maximal degree in the table: $D = 12$ and the size of the matrix is 2^{46} so the total cost is bigger than $2^{46\omega} \geq 2^{92}$. In [BFS03] we give explicit expressions for N_D in function of D and n; the two following formulas are useful for HFE:

Proposition 5 *Let S be a system of n random equations in n variables. During the computation of S with F_5 the size of matrix at degree D is:*

Number of rows	*in degree 4*	*in degree 5*
in the matrix	$1/2\, n\,(1+n^2)$	$1/6\, n\,(n-1)\,(n-3)\,(n+1)$

4 Experimental results

The results of this section are all coming from experiments: we are running Gröbner basis computations for real size HFE problems; then we analyse the results in the light of theoretical results obtained in section 3.4.

Let HFE(d, n) be the algebraic system corresponding to the basic HFE problem with $f(x)$ a random (quadratic) polynomial of degree d and random coefficients in the field $\mathbb{F}_{2^n}$.

To generate the system of equations HFE(d, n) we have used two programs: one written by JF Michon using NTL[Sho03] the other one being written in C by D. Augot. For instance when $d = 16$ and $n = 12$ it takes 1 min 25 sec (PIII 1000 Mhz) to generate the algebraic system and the size of the output file is 13 Mbytes.

From the Gröbner point of view we note that the two affine transform (see section 2) S and T are useless: the effect T (resp. S) is equivalent to a random change of coordinates (resp. to replace the generators of the ideal I by linear combinations). Hence the ideal (the hilbert function) remains unchanged. Of course, without S, T, HFE could be attacked by *other* methods.

4.1 First HFE Challenge is broken

The first HFE Challenge was proposed in [Pat96c] with a (symbolic) prize of 500\$. This correspond to a HFE$(d = 96, n = 80)$ problem and can be downloaded from [Pat96a]. For this problem, the exhaustive search attack require $\geq 2^{80}$ operations, hence is not feasible.

We have computed a Gröbner basis of this system with the algorithm F_5 (in fact a special version for $\mathbb{F}_2$) implemented in the FGb (Fast Gb) Gröbner program (written in C). As explained in section 3.3 the most time consuming part is linear algebra: for this example we have solve a 307126×1667009 matrix over $\mathbb{F}_2$. The total running time was 187892 sec ($\approx$ 2 days and 4 hours) on an HP workstation with an alpha EV68 processor at 1000 Mhz

and 4Go bytes of RAM. Some care has been taken for the memory management since the size of the process was 7.65 Giga bytes.

For this algebraic systems [Pat96a] we found that there were *four solutions*:

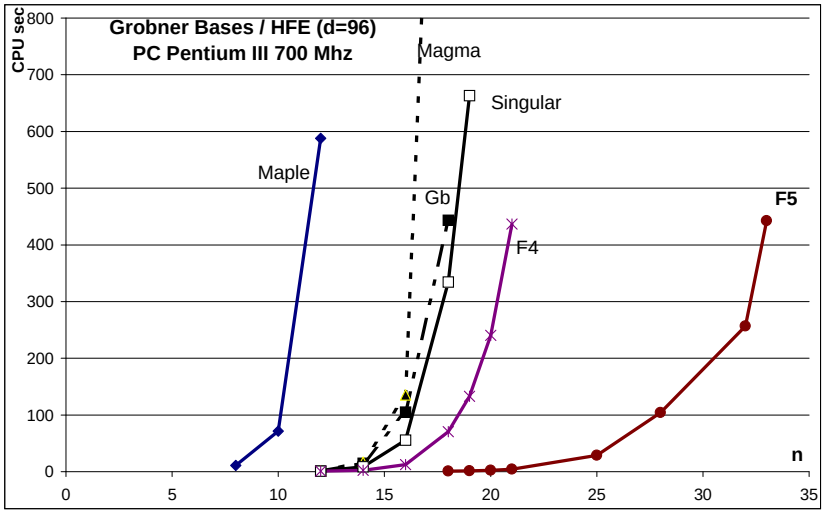
$$X = 644318005239051140554718 \quad X = 934344890045941098615214$$
$$X = 102267771362902876120304 6 \quad X = 103704608265180114959467 0$$

where $X = \sum_{i=1}^{80} x_i 2^{i-1}$.

It must be emphasized that this computation is far beyond the capacity of all the other implementations and algorithms for computing Gröbner basis as is made clear by the following table:

	Algo	10	12	14	16	18	19	21	33
Maple	(Buchberger)	71.7 s	587.9 s						
Magma	(Buchberger)		1.5 s	17.0 s	135.4 s	1900 s			
Gb	(Buchberger)		0.8 s	15.1 s	105 s	443.2 s			
Singular	(Buchberger)		0.7 s	8.6 s	55.5 s	334.3 s	663.4 s		
FGb	F_4			2.4 s	12.3 s	70.5 s	133.2 s	436.9 s	
FGb	$F_5/2$					**0.9 s**	**1.5 s**	**4.25 s**	**442.7 s**

Comparison of various algorithms and implementations (PC PIII 700 Mhz)

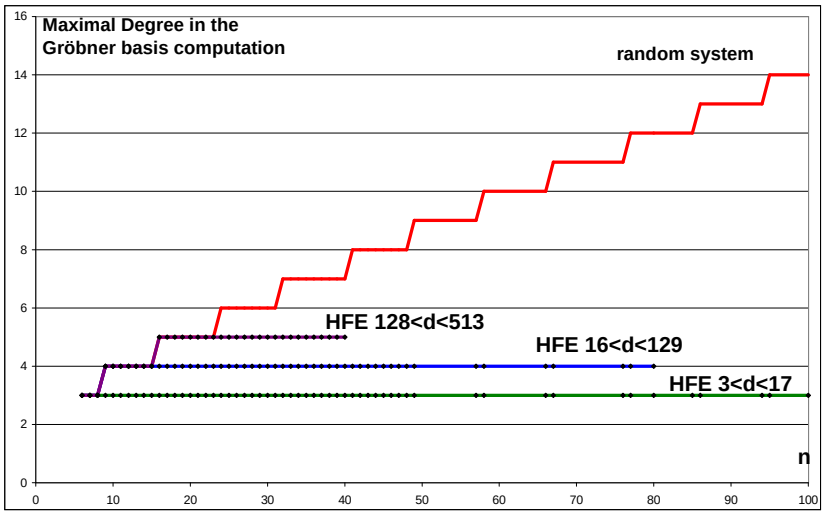


Comparison of various algorithms and implementations (PC PIII 700 Mhz)

Because 80 equations of degree 2 was a previously untractable problem, this Gröbner computation represents a breakthrough in research on polynomial system solving.

4.2 HFE algebraic systems are not random

We have collected a lot of experimental data by running thousand of HFE systems for various $d \leq 1024$ and $n \leq 160$. In the following graph, the maximal degree occurring in the Gröbner basis computation of an algebraic system coming from HFE (resp. from a random system as described in table 1 section 3.4) is plotted:



Small dots correspond to a computer simulation.

As is made clear by this graph, HFE algebraic system are not equivalent to random system from the Gröbner basis point of view.

Remark 2 *A common pitfall is to compare an HFE algebraic system and random system for too small values of* n*. For instance, if we want to* experimentally prove *that the maximal degree occurring in the computation of* $HFE(129, n)$ *is always less than* 5*. We read in table 1 (section 3.4) that we must take* $n \geq 24$*: in fact when* $n < 24$ *for all random systems in* n *variables the computation stops at degree* 5*. Hence, when* $n < 24$ *it is impossible to distinguish* $HFE(129, n)$ *from a random system.*

Remark 3 *For the first HFE challenge, the difference with a random system can be detected after* 6 *hours of computation.*

From the graph and [KS99, Cou01] it is natural to conjecture:

Proposition 6 *The basic HFE problem corresponding a secret polynomial of degree d with coefficients in the field $\mathbb{F}_{2^n}$ can be solved in $\mathcal{O}(n^{\omega D})$ where $D < \log_2(d)$ is the maximal degree occurring in the computation. For practical value of $d < 513$ we have $D \leq 5$. More precisely $D \leq 4$ (resp. $D \leq 3$) when $D \leq 128$ (resp. $D \leq 16$).*

4.3 Experimental complexity

We know from proposition 6 that the complexity for computing a Gröbner basis of HFE(d, n) is polynomial in n (say $\mathcal{O}(n^{k_d})$) when d is fixed but we want to find precisely k_d for practical value of $d < 513$. Consequently we analyse some simulation results. From a practical point of view the complexity could be the running-time but it is a noisy measurement: it depends strongly on the architecture (32 or 64 bits), the size of the various level of cache, the load of the computer, ... Hence we give *also* the total number of arithmetic operations: since the most consuming part is linear algebra over $\mathbb{F}_2$ we give the number of 64 bits xor operations (XOR). This number is the same for all computers and depends only on the linear algebra that we have implemented (in our case standard Gaussian elimination).

First we want to establish that there are only three "class of complexity" when $d < 513$:

C_1 when $4 < d < 17$ all the HFE(d, n) are in roughly equivalent.

C_2 when $16 < d < 129$ all the HFE(d, n) are in roughly equivalent.

C_3 when $128 < d < 513$ all the HFE(d, n) are in roughly equivalent.

For all admissible values of d inside a class C_j we compare $HFE(d, n)$ with a "reference degree": 12 (class C_1), 17 (C_2), 129 (C_3). For instance for the second class C_2 we compare HFE(d, n) with HFE$(17, n)$:

n	21	22	23	24	28	30	32	33	40	41	49	50	60	64	70	80
XOR(96)/XOR(17)	5.3	6.4	7.3	8.3	9.4	9.3	9.1	9.0	8.3	8.2	7.6	7.5	6.9	6.7	6.4	6.1
CPU(96)/CPU(17)	3.8	4.6	5.2	5.9	6.5	6.4	6.2	6.1	5.7	5.6	5.1	5.0	4.7	4.9	5.1	4.4

Comparison: HFE(96,n)/HFE(17,n)

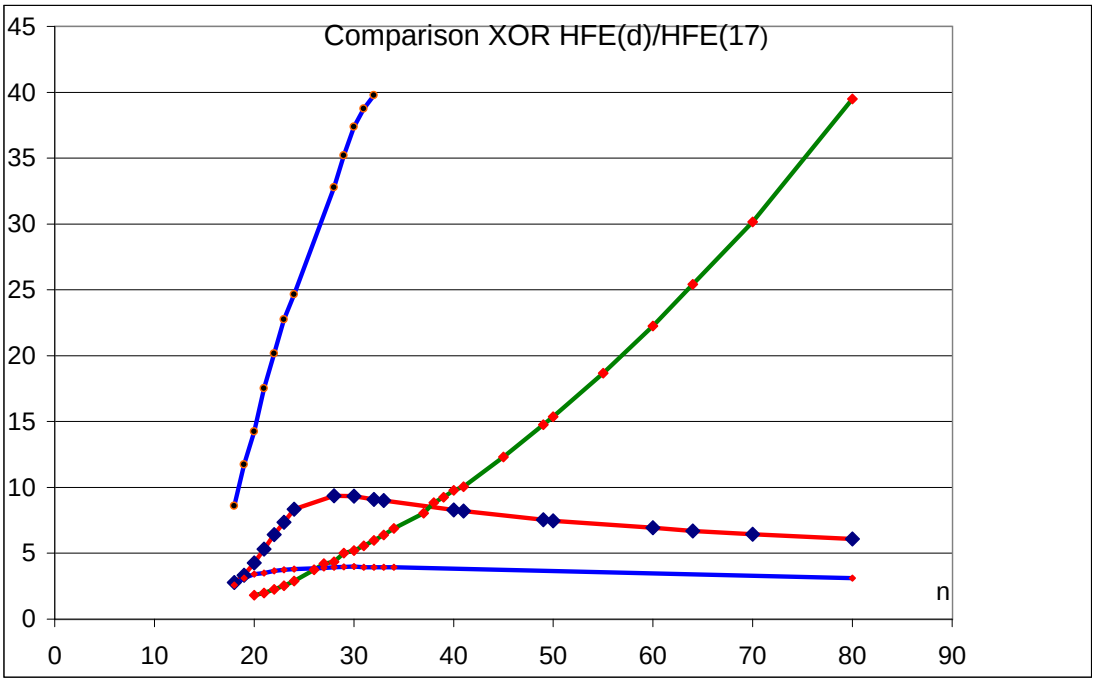
From this table we conclude that HFE$(96, n)$ is only 6 times more difficult than HFE$(17, n)$ so that the exponents k_{96} and k_{17} are the same. On the other the hand the following results clearly indicates that $k_{12} < k_{17} < k_{129}$:

n	40	41	45	49	50	55	60	64	70	80
XOR(17)/XOR(12)	195.6	200.9	246.4	295.2	307.3	373.6	444.9	508.3	603.2	789.7
CPU(17)/CPU(12)	131.4	133.7	182.1	251.6	262.0	374.3	487.6	701.3	932.4	1505.4

Comparison: HFE(17,n)/HFE(12,n)

n	18	19	20	21	22	23	24	28	29	30	31	32
XOR(129)/XOR(17)	21.5	29.3	35.6	43.8	50.4	56.9	61.6	81.9	88.0	93.4	96.9	99.4
CPU(129)/CPU(17)	35.8	49.5	61.2	80.0	100.2	125.0	145.5	246.5	292.1	329.7	360.7	397.3

Comparison: HFE(129,n)/HFE(17,n)



Small dots correspond to a computer simulation.

To sum up:

$$\text{HFE(129,n)} \gg \text{HFE(96,n)} \approx \text{HFE(17,n)} \gg \text{HFE(12,n)}$$

Next we want to find k_{12}, k_{17}, k_{129}. We have used several methods to find the exponent. We begin by a theoretical analysis made on the following hypothesis: suppose that the HFE(d,n) behave like a random system except that the maximal degree occurring in the computation D' is much less than D (given by the bound of section 3.4). If this is true we

have to solve a linear system whose size is $r \times c$ where r is given by proposition 5 and c the number of columns is simply the number of monomials in degree D': $c = \sum_{i=1}^{D'} \binom{n}{i}$.

Since the rows of the matrix have the shape $m\, f_i$ and f_i is a polynomial of degree 2, the number of non zero elements in the $r \times c$ matrix is at most $\mathrm{NZ} = r\,\frac{n\,(n+1)}{2}$. By applying sparse linear algebra technique (Wiedemann's algorithm, ...), we can find the solution in $\mathcal{O}(r\mathrm{NZ}) = \mathcal{O}(r^2 n^2)$ operations. From proposition 6 we know that $D' \leq 4$ when $d \leq 128$ and in that case $r \approx \frac{n^3}{2}$ (from proposition 5). Consequently the total cost is $\mathcal{O}(n^8)$. In the same way we found $\mathcal{O}(n^{10})$ (resp. $\mathcal{O}(n^6)$) when $129 \leq d \leq 512$ (resp $d < 17$).

It must be emphasized that the previous computation is not a complexity proof since we cannot check the hypothesis. We must confirmed this results experimentally by doing *real simulations*.

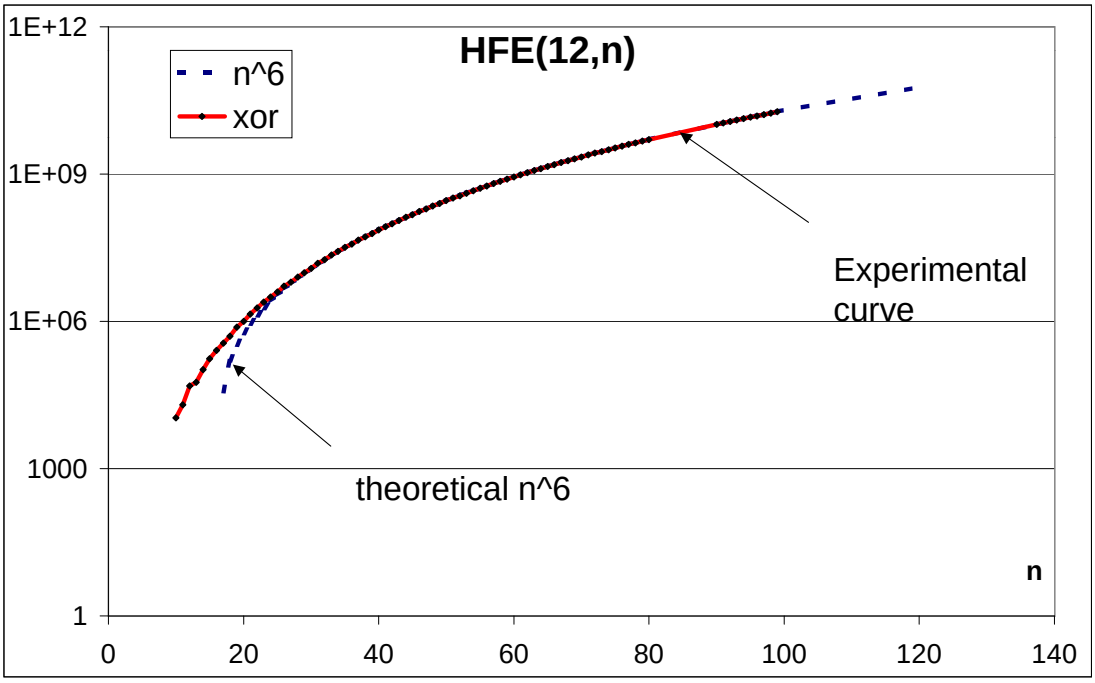
Suppose that the complexity is $f(n) = Cn^k$ then we draw the curve $\log(f(n)) = \log(C) + k\log(n)$ to find the slope of line. We can also try to find a good polynomial approximation (using least squares method) of the curve $f(n)$. For instance, when $\mathrm{HFE}(d = 12, n)$ we have collected a set of data $10 < n < 160$ and found:

$$\frac{f(n+1)}{f(n)} \approx 1 + \frac{6.129055587}{n}$$
$$log(f(n)) \approx 6.086191079\, log(n) - 4.324402957$$
$$f(n) \approx -194.0797\, n^3 + 11.1747\, n^4 - .3354\, n^5 + .02212\, n^6$$
$$f(n) \approx 224.4411\, n^3 - 15.4212\, n^4 + .2696\, n^5 + .01623\, n^6 + 2.0810^{-5}\, n^7$$

Hence it is clear that $k_{12} = 6$. We report in the following tables the result of our simulations. The running-times are given for HP workstation with an alpha EV68 processor at 1000 Mhz. (C_1 and C_2 are constants):

n	93	94	95	96	97	98	99	100	120	140	160
XOR	$2^{33.6}$	$2^{33.7}$	$2^{33.7}$	$2^{33.8}$	$2^{33.9}$	$2^{34.0}$	$2^{34.1}$	$2^{34.2}$	$2^{35.8}$	$2^{37.1}$	$2^{38.3}$
$C_1 n^6$	$2^{33.6}$	$2^{33.7}$	$2^{33.7}$	$2^{33.8}$	$2^{33.9}$	$2^{34.0}$	$2^{34.1}$	$2^{34.2}$	$2^{35.8}$	$2^{37.1}$	$2^{38.3}$
$\mathrm{XOR}/C_1 n^6$	.998	1.000	1.001	1.000	1.000	1.002	1.001	.999	1.001	1.000	1.000
CPU (sec)	$2^{6.2}$	$2^{6.2}$	$2^{6.3}$	$2^{6.4}$	$2^{6.5}$	$2^{6.6}$	$2^{6.6}$	$2^{6.7}$	$2^{8.4}$	$2^{9.8}$	$2^{10.9}$
$C_2 n^6$	$2^{6.1}$	$2^{6.2}$	$2^{6.3}$	$2^{6.4}$	$2^{6.5}$	$2^{6.6}$	$2^{6.7}$	$2^{6.8}$	$2^{8.4}$	$2^{9.8}$	$2^{10.9}$
$\mathrm{CPU}/C_2 n^6$	1.047	.996	.987	.984	.978	.992	.963	.944	1.033	.995	1.000

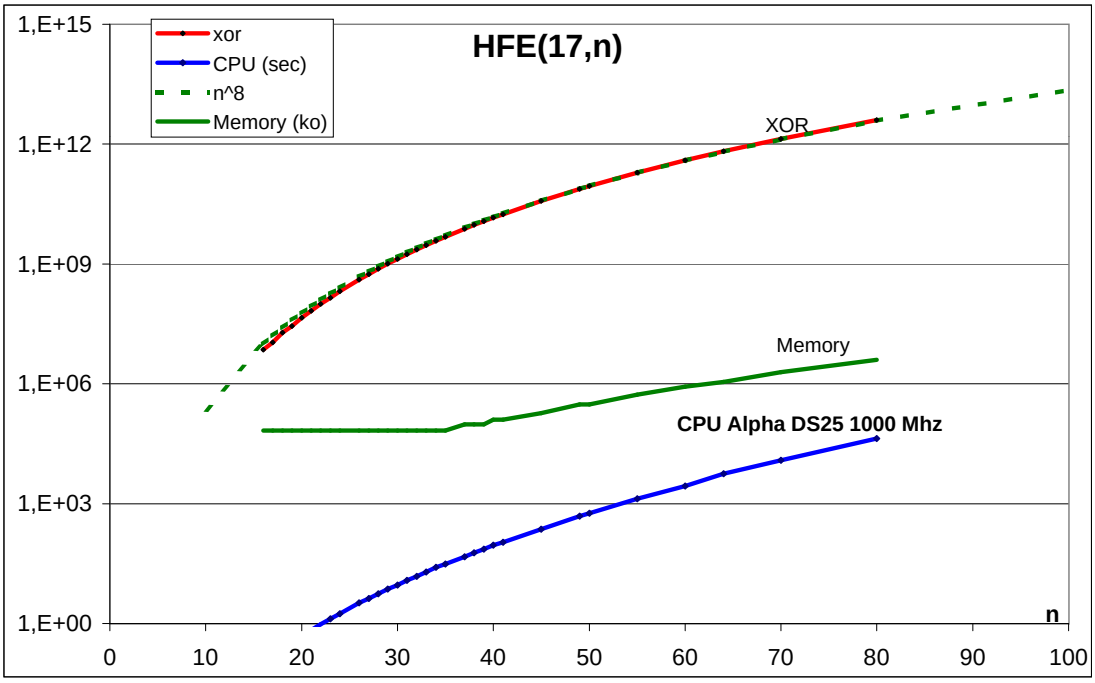
Comparison between running-times for $\mathrm{HFE}(12, n)$ and theoretical $\mathcal{O}(n^6)$



Comparison between running-times for HFE$(12, n)$ and theoretical $\mathcal{O}(n^6)$

n	41	45	49	50	55	60	64	70	80
XOR	$2^{34.0}$	$2^{35.1}$	$2^{36.1}$	$2^{36.4}$	$2^{37.5}$	$2^{38.5}$	$2^{39.3}$	$2^{40.3}$	$2^{41.9}$
$C_1 n^8$	$2^{34.0}$	$2^{35.1}$	$2^{36.1}$	$2^{36.4}$	$2^{37.5}$	$2^{38.5}$	$2^{39.3}$	$2^{40.3}$	$2^{41.9}$
XOR$/C_1 n^8$)	1.014	1.003	.999	.999	.999	.997	1.002	1.000	1.000
CPU (sec)	$2^{6.8}$	$2^{7.8}$	$2^{8.9}$	$2^{9.2}$	$2^{10.4}$	$2^{11.4}$	$2^{12.5}$	$2^{13.6}$	$2^{15.4}$
$C_2 n^8$	$2^{6.5}$	$2^{7.7}$	$2^{8.9}$	$2^{9.1}$	$2^{10.4}$	$2^{11.5}$	$2^{12.4}$	$2^{13.6}$	$2^{15.4}$
CPU$/C_2 n^8$	1.159	1.092	1.054	1.026	.991	.927	1.045	.995	1.000

Comparison between running-times for HFE$(17, n)$ and theoretical $\mathcal{O}(n^8)$



Comparison between running-times for HFE$(17, n)$ and theoretical $\mathcal{O}(n^8)$

All results presented in the above tables (and similar simulations for other values of d) confirm the validity and the accuracy of the previous estimation.

Theorem 1 *The complexity of the Gröbner basis compute HFE(d, n) is:*

degree of $f(x)$	$d \leq 16$	$17 \leq d \leq 128$	$129 \leq d \leq 512$
Gröbner complexity	$\mathcal{O}(n^6)$	$\mathcal{O}(n^8)$	$\mathcal{O}(n^{10})$

5 Patarin original attack revisited

It is interesting to compare the Gröbner bases method with the original attack of Patarin [Pat95a]. We consider the "toy example" ([MI88] page 420): the secret key is $f(x) = x^3$, $n = 8$ that is to say the field is $\mathbb{F}_{2^8}$ and the public key is:

$[x_0 + x_1 + x_3 + x_7 + x_0\,x_1 + x_0\,x_2 + x_0\,x_4 + x_0\,x_5 + x_0\,x_6 + x_0\,x_7 + x_1\,x_4 + x_1\,x_6 + x_1\,x_7 + x_2\,x_6 + x_3\,x_4 + x_3\,x_5 + x_3\,x_7 + x_4\,x_5 + x_5\,x_6 + x_5\,x_7 + y_0,$
$x_1 + x_2 + x_4 + x_6 + x_0\,x_3 + x_0\,x_6 + x_0\,x_7 + x_1\,x_3 + x_1\,x_4 + x_1\,x_6 + x_2\,x_5 + x_2\,x_7 + x_3\,x_4 + x_3\,x_7 + x_4\,x_6 + x_4\,x_7 + x_6\,x_7 + y_1,$
$1 + x_0 + x_1 + x_2 + x_3 + x_5 + x_6 + x_0\,x_1 + x_0\,x_2 + x_0\,x_5 + x_1\,x_2 + x_1\,x_4 + x_1\,x_6 + x_1\,x_7 + x_2\,x_6 + x_2\,x_7 + x_3\,x_5 + x_3\,x_6 + x_3\,x_7 + x_4\,x_5 + x_5\,x_6 + x_5\,x_7 + y_2,$
$x_0 + x_2 + x_3 + x_7 + x_0\,x_3 + x_0\,x_5 + x_1\,x_4 + x_1\,x_5 + x_1\,x_6 + x_1\,x_7 + x_2\,x_3 + x_2\,x_4 + x_2\,x_7 + x_3\,x_4 + x_3\,x_7 + y_3,$
$1 + x_0 + x_1 + x_2 + x_6 + x_7 + x_0\,x_2 + x_0\,x_4 + x_0\,x_5 + x_1\,x_3 + x_1\,x_7 + x_2\,x_6 + x_3\,x_4 + x_3\,x_5 + x_3\,x_6 + x_4\,x_5 + x_4\,x_6 + x_4\,x_7 + x_5\,x_6 + x_5\,x_7 + x_6\,x_7 + y_4,$
$x_4 + x_6 + x_0\,x_1 + x_0\,x_2 + x_0\,x_3 + x_1\,x_2 + x_1\,x_3 + x_1\,x_5 + x_2\,x_6 + x_3\,x_4 + x_3\,x_7 + y_5,$
$1 + x_0 + x_2 + x_3 + x_7 + x_0\,x_1 + x_0\,x_4 + x_1\,x_3 + x_1\,x_4 + x_1\,x_6 + x_2\,x_3 + x_2\,x_4 + x_2\,x_6 + x_3\,x_4 + x_3\,x_5 + x_3\,x_7 + x_4\,x_5 + x_4\,x_6 + x_4\,x_7 + x_5\,x_6 + x_5\,x_7 + x_6\,x_7 + y_6,$
$x_0 + x_1 + x_4 + x_5 + x_7 + x_0\,x_1 + x_1\,x_3 + x_1\,x_5 + x_2\,x_6 + x_2\,x_7 + x_3\,x_5 + x_3\,x_6 + x_3\,x_7 + x_4\,x_5 + x_5\,x_6 + x_5\,x_7 + y_7]$

5.1 Patarin's attack

The idea of Patarin (see [Pat95b] page 12) is to find a low degree relation between x and $y = f(x)$. In our case we have, for instance, $y^5 = (x^3)^5 = x^{15} = x$. This imply that we can find equations of degree 2 (since $5 = 1 + 4$) in y and 1 in x:

$$a_0 + \sum_{i=0}^{7} b_i x_i + \sum_{0 \le i < j \le 7} c_{i,j} y_i y_j + \sum_{i=0}^{7} d_i y_i = 0$$

This give three independent equations:

$$\begin{aligned}
x_0 + x_1 + x_3 + \quad & x_4 + x_7 = y_0\,y_2 + y_0\,y_3 + y_2\,y_4 + y_3\,y_4 + y_0\,y_7 \\
& + y_4\,y_7 + y_2 + y_4 + y_5 + y_6 + y_7 \\
x_2 + x_3 = \quad & y_2\,y_3 + y_2\,y_4 + y_3\,y_4 + y_2\,y_5 + y_3\,y_5 + y_2\,y_6 + y_3\,y_6 \\
& + y_3\,y_7 + y_4\,y_7 + y_5\,y_7 + y_6\,y_7 + y_0 + y_2 + y_3 + y_5 + y_6 + y_7 \\
x_1 + x_6 + x_7 = \quad & y_0\,y_3 + y_0\,y_4 + y_3\,y_4 + y_0\,y_5 + y_4\,y_5 + y_0\,y_6 + y_4\,y_6 \\
& + y_2 + y_3 + y_4 + y_7 + 1
\end{aligned}$$

As a result, from these equations, we can eliminate three variables x_0, x_1 and x_2 in the 8 public equations. We obtain 5 equations of degree 2. We can now find the solution by doing an exhaustive search for 3 variables (x_3, x_4 and x_5).

5.2 Generic Gröbner bases in precomputation phase

Proposition 2 and 3 tell us that a computation of the Gröbner basis (for an appropriate ordering) of the public equations give us the relations among the y_i of *lowest degree*. The Gröbner bases G contains 178 polynomials; among them 24 are of total degree two and linear in x_i (we will denote G' this subset of G). G' contains the three previous linear equations found in [Pat95b] but also many others; for instance one such equation is:

$$x_3 + x_2 + (x_1 + x_6 + x_7)(y_4 + y_3 + y_6 + y_5 + 1) + y_0 + y_4 + 1 = 0$$

Now to find the solution it is enough to substitute the values of y_i (can be done in $\mathcal{O}(n^3)$ operations) and then to solve a linear system (again $\mathcal{O}(n^3)$ operations). So the most costly operation is the computation of G': since G' contains equations of degree 2 this can be done (by linear algebra techniques) in $\mathcal{O}(n^7)$ operations. Note that this step has to be done only once (one precomputation phase for each new public key); hence the complexity is $\mathcal{O}(n^7) + K\mathcal{O}(n^3)$ where K is the number of messages to decipher. This must be compared with the complexity $\mathcal{O}(n^6)$ found in section 4.3 (theorem 1). We conclude that Gröbner bases is useful to find *automatically* all the low degree relations; the drawback is that we cannot find a bound similar to [Pat95b] for the number of independent equations.

6 Conclusion

We have presented a very efficient attack on the basic HFE cryptosystem based on Gröbner bases computation. It is not only a theoretical attack with a good complexity but also a very practical method since our implementation was able to break the first HFE challenge (80 bits). However, several modified versions of HFE have been proposed [Pat96b, PGC98]. These perturbations (for instance one can simply remove some equations of the public key) are applied to the basic HFE and are expected to make attacks harder. Hence, the HFEv- is the modified version of HFE used in Quartz that has been submitted to European project NESSIE. It is an open issue to evaluate the practical robustness of these modified versions of HFE by using the techniques presented in this paper.

Acknowledgements

We would like to thank the JF Michon and D Augot for their programs. We gratefully acknowledge several useful discussions with J. Patarin who also introduced to him his work on HFE. I am indebted to the LIP6 for its partial support of this work (Alpha DS25).

References

[Bec93] Becker T. and Weispfenning V. *Groebner Bases, a Computationnal Approach to Commutative Algebra.* Graduate Texts in Mathematics. Springer-Verlag, 1993.

[BFS03] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. Complexity of gröbner bases computation of generic systems. in preparation, 2003.

[Buc65] Buchberger B. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* PhD thesis, Innsbruck, 1965.

[Buc70] Buchberger B. An Algorithmical Criterion for the Solvability of Algebraic Systems. *Aequationes Mathematicae*, 4(3):374–383, 1970. (German).

[Buc79] Buchberger B. A Criterion for Detecting Unnecessary Reductions in the Construction of Gröbner Basis. In *Proc. EUROSAM 79*, volume 72 of *Lect. Notes in Comp. Sci.*, pages 3–21. Springer Verlag, 1979.

[CLO92] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties and Algorithms.* Springer Verlag, New York, 1992.

[Cou01] Nicolas T. Courtois. The security of hidden field equations (hfe). In *Cryptographers' Track RSA Conference*, volume 2020 of *Lecutures Notes in Computer Science*, pages 266–281, 2001.

[CSPK00] Nicolas Courtois, Adi Shamir, Jacques Patarin, and A. Klimov. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Eurocrypt'2000*, volume 1807 of *Lecutures Notes in Computer Science*, pages 392–407. Springer Verlag, 2000.

[Fau99] Faugère J.C. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1–3):61–88, June 1999.

[Fau02] Faugère J.C. A new efficient algorithm for computing Gröbner bases without reduction to zero F5. In T. Mora, editor, *Proceedings of ISSAC*, pages 75–83. ACM Press, July 2002.

[KS99] Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. 1666:19–30, 1999.

[Laz83] Lazard D. Gaussian Elimination and Resolution of Systems of Algebraic Equations. In *Proc. EUROCAL 83*, volume 162 of *Lect. Notes in Comp. Sci*, pages 146–157, 1983.

[Mac16] F.S. Macaulay. *The algebraic theory of modular systems.*, volume xxxi of *Cambridge Mathematical Library.* Cambridge University Press, 1916.

[MI88] Tsutomu Matsumoto and Hideki Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In *Crypto 88*, volume 330 of *Lecutures Notes in Computer Science*, page 419. Springer Verlag, 1988.

[Pat95a] Jacques Patarin. Cryptanalysis of the matsumoto and imai public key scheme of eurocrypt'88. In *Proc. of the 15th Annual International Cryptology Conference on Advances in Cryptology - CRYPTO'95*, pages 248–261, Santa Barbara, California, 1995.

[Pat95b] Jacques Patarin. Cryptanalysis of the matsumoto and imai public key scheme of eurocrypt'88. Extended version, 1995.

[Pat96a] Jacques Patarin. *HFE first challenge*, 1996. http://www.minrank.org/challenge1.txt.

[Pat96b] Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In *EUROCRYPT'96*, volume 1070 of *Lecutures Notes in Computer Science*, pages 33–??, 1996.

[Pat96c] Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. Extended version, 1996.

[PGC98] Jacques Patarin, Louis Goubin, and Nicolas Courtois. Improved algorithms for isomorphisms of polynomials. 1403:184–??, 1998.

[Sho03] V. Shoup. *NTL 5.3.1, a Library for doing Number Theory*, 2003. `http://www.shoup.net/ntl`.

[vzGG99] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Cambridge Press, 1999.

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Futurs : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399